Sprint 2 Plan - EasyTrage, TeamTrage

Sprint completion date 2/18/18

Revision 2, Revision Date 2/15/18

Red = unmet goal, blue = revision

Goals: Finish user registration and login, display prices, trading fees, current transaction time and cost, etc for different exchanges of at least 1 coin and display arbitrage suggestions between those exchanges for that coin.

Tasks:

1. (5 story points) User story 0: As an uninitiated user, I want to be able to successfully create an account on EasyTrage and have my data persist in a database
   a. Change username on login form to be email, and validate email/pw based on common requirements (ie pw is 6+ letters, email has @ in it, etc) (1-2 hour)
   b. Upon submit, disable form. If form data is valid, send a POST/PUT request to the backend server with email and hashed password (use bcrypt.js) (1-2 hours)
   c. Write backend function(s) to check accuracy of user details entered against db / make new db entry for new user (2-4 hours)

2. (8 story points) User story 1: As an EasyTrage user, I want to see coin prices, trade fees, current estimated transaction time, estimated transaction fees for coins from various exchanges
   a. Decide on a coin, exchanges as a team and make API requests on the backend to store coin prices across different exchanges on the server every x seconds (2-4 hours)
   b. Look for API to check current transaction cost / time and call it to store that info on the server as well, and request/hardcode exchange fees (1-2 hours)
   c. ~~Write react front-end component to request display this information in a table (and add charts / visual representation if time permits) on the dashboard (2-4 hours)~~

3. (8 story points) User story 2: As an EasyTrage user, I want to see trade suggestions for at least 1 of coin
   a. Use pulled data in server to calculate if there are any profitable arbitrage opportunities and store them in a sorted list (based on profitability %) of trades on the server (4-6 hours)
   b. Write react front-end component for "trade suggestions" section of dashboard which requests backend server for the array of suggestions (2-4 hours)

4. (3 points) User story 3: As a developer, I want to separate my client, server, and database processes into different docker containers update them automatically when I make changes without rebuilding.
   a. mount containers to local folders using docker volumes so that we don't need to rebuild and the apps refresh when we make changes (1-2 hours)

      b. Make 3 different containers using docker compose for server, client, mongo so that only one refreshes when changes are made to only one of them. (1 hour)

Roles:
- <u>Mike Hamilton</u>: Developer
- <u>Christopher Kong</u>: Developer
- <u>Kevin Lee</u>: Developer, Scrum Master
- <u>Vikram Melkote</u>: Product Owner, Developer
- <u>Abdul Mohamed</u>: Developer

Task Assignment:
- <u>Mike Hamilton</u>: User Story 0: c, User Story 1: a, b, User Story 2: a
- <u>Christopher Kong</u>: User Story 0: a, b, User Story 1: c, User Story 2: b
- <u>Kevin Lee</u>: User Story 0: a,b User Story 1: c, User Story 2: b
- <u>Vikram Melkote</u>: User Story 0: c, User Story 1: Tasks a and b, User Story 2: a, User Story 3: a
- <u>Abdul Mohamed</u>: User Story 1: all, User Story 2: a, User Story 3: a

Burnup Chart: